

Librería Motor pasos a paso (MPaP) UNIPOLARES

Francisco Javier Rodríguez Navarro
www.pinguytaz.net

Indice

1.Introducción.....	1
1.1.Motores.....	1
2.Motores paso a paso.....	3
2.1.Características de interés.....	4
a)Frecuencia de paso máximo Pull-in/out.....	5
b)Ángulo de paso.....	5
c)Pasos por vuelta.....	5
d)Alimentación.....	5
2.2.Drivers de motor.....	6
2.3. Como conocer los terminales.....	8
3.Librería MpaP.....	11
3.1. Constructores.....	11
3.2. Variables.....	13
3.3.Métodos.....	15
4.Características motor.....	17
4.1.Unipolares.....	17
5.Instalación.....	18
6.Glosario.....	20

1.Introducción

1. INTRODUCCIÓN

Esta librería nos permite utilizar motores paso a paso “Unipolares” con solo indicar los pines, tipo de movimiento y velocidad nos permitirá movernos por pasos o por ángulos.

1.1. Motores

Los motores son los dispositivos que dan movilidad, ya sea a un robot o a cualquier otra aplicación que se nos ocurra. Tenemos varios tipos de motores:

- **Corriente continua** y son los que en aplicaciones como robótica suelen dar el giro de las ruedas, suelen tener gran estabilidad de giro pero es difícil controlar el ángulo girado debiéndose usar elementos externos. En los motores de continua la velocidad es directamente proporcional a la tensión aplicada, para controlar la velocidad se pueden usar varios métodos pero el más común es el de PWM (No obstante esto no es objeto de esta librería)
- **Servomotores** Similar a los de corriente

1.Introducción

continua pero nos permite controlar el ángulo (dentro de un rango) y estar estable en esa posición. Estos motores se controla mediante salidas PWM.

- **Paso a paso** Estos nos permiten conocer el ángulo de giro ya que como su nombre indica gira un ángulo determinado por cada paso que le marcamos, podrán ser Unipolares y Bipolares y ya que es objeto de esta librería les dedicamos un apartado. Estos motores los controlaremos mediante salidas digitales con secuencias programadas.

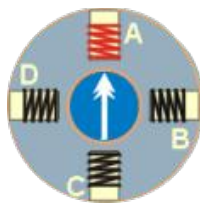
2. Motores paso a paso

2. MOTORES PASO A PASO

Aunque nos podemos encontrar varios tipos de motores paso a paso como son los de redundancia variable, imán permanente e híbridos, nos centraremos solo en los de imán permanente y dentro de ellos en los Unipolares.

El movimiento de estos motores se logra gracias a las fuerzas de atracción y repulsión magnética que se logra con dos bobinas:

- Estátor: bobina generalmente circular y fija. Por lo general dos bobinados y cuatro polos.
- Rotor: Es la bobina localizada en el centro del estátor, que es móvil y busca el equilibrio según la polaridad que se aplique a las bobinas del estátor.



2.Motores paso a paso

De esta forma visto giraríamos en pasos de 90° ya que esto no sería práctico se introducen reductoras que nos permiten pasos más pequeños, podríamos utilizar más bobinas pero esto incrementaría los costes.

Los motores unipolares tendrán 5 ó 6 cable y 4 de ellos corresponderán son los terminales de la bobina que se alimentaran mediante un driver motor (del que hablaremos más adelante) para darle la suficiente corriente pues la que nos proporciona el arduino no es suficiente.

2.1. Características de interés

A continuación damos las características más interesantes de un motor paso a paso, en especial para configurar la librería, pero existen otras como momento de inercia, par motor, etc. que será necesario conocer para la parte mecánica de nuestros proyectos.

a) Frecuencia de paso máximo Pull-in/out

Nos indica el número máximo de pasos por se-

2.Motores paso a paso

gundo que puede dar el motor sin que el motor deje de responder, en nuestra librería se definirá en la variable *pps*.

Por regla general el fabricante dará curvas Pull-in/Out.

Con este parámetro podremos calcular la velocidad máxima que puede realizar el motor.

b) Ángulo de paso

Los grados que se avanza con cada paso, impulso, se mide lógicamente en grados y con este valor podremos calcular el número de pasos totales para una vuelta con una sencilla formula " $360/\text{angulo}$ ".

c) Pasos por vuelta

Complementario al anterior nos da el número de pasos que se tiene que dar para generar una vuelta, y al igual que antes con este valor podremos calcular el ángulo que gira usando la formula " $360/\text{pasos}$ ".

d) Alimentación

Datos de corriente, voltaje y resistencia. Estos parámetros son importantes para conocer como alimentaremos al motor.

2.Motores paso a paso

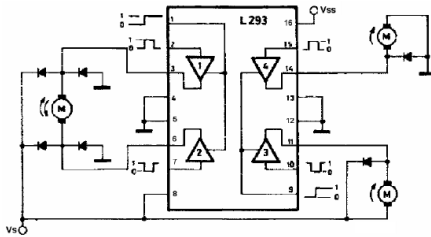
2.2. Drivers de motor

El driver de motor es un circuito que nos permite proporcionar la corriente necesaria a los motores ya que nuestro arduino no podría. Como ejemplo para el uso de esta librería se propone el uso del L293 pero existen muchos otros modos e incluso circuitos ya realizados.

El L293 es un doble puente H que se encuentra en un integrado y nos permite, como se ha comentado antes, proporcionar la corriente necesaria ya que puede tener picos de hasta 1A y a la solicitud de 5V que dará un arduino podremos generar una salida de hasta 1A y la tensión de salida puede ser de hasta 36V.

Tenemos dos L293 uno con la letra "D" y otro "B", la diferencia es que el "B" va sin diodos de protección (protege al integrado de la tensión de inducción de los motores) y los deberemos poner nosotros y tendremos un esquema similar al siguiente

2.Motores paso a paso



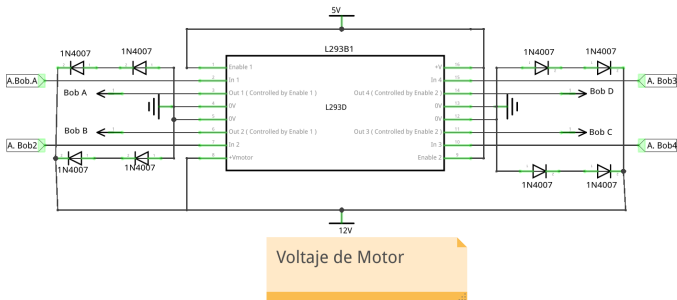
Vi (cada canal)	Vo	Vinh
H	H	H
L	L	H
H	Z	L
L	Z	L

En nuestro caso las dos entradas de de habilitación las pondremos en alto, ya que solo queremos activar las bobinas y no generar pulsos PWM.

El circuito del Driver motor que hemos construido quedaría de la siguiente forma, pero recordar que se puede generar cualquier otro Driver motor con 4 entrada (Pines al Arduino) y 4 salidas (Pines al motor)

Libreria Motor pasos a paso (MPaP) UNIPOLARES

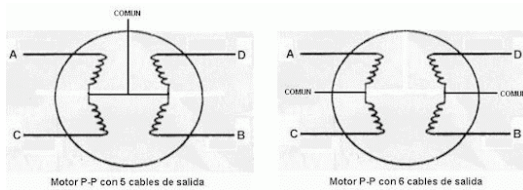
2. Motores paso a paso



fritzing

2.3. Como conocer los terminales

Si no disponemos de la identificación de los terminales del motor, vamos a explicar como descubrirlos.



Lo primero mediremos con un polímetro todos los terminales para ver cuales son los terminales comunes.

1) Si no hay continuidad, esto podrá pasar

Libreria Motor pasos a paso (MPaP) UNIPOLARES

2.Motores paso a paso

en 6 terminales, es que pertenecen a dos grupos diferentes.

- 2) Si nos da el doble de resistencia esperada, es que estamos midiendo con dos bobinas.
- 3) Si es la resistencia esperada, uno de ellos sera común y el otro bobina, esta claro que el común tendrá al menos otro terminal que le de ese resultado.

Con los comunes localizados los conectaremos a masa y ahora procederemos de la siguiente forma.

1. Vamos conectando los otros de uno en uno a positivo y uno de ellos se moverá en el sentido de las manecillas de reloj (Llamaremos bobina A) y otro al contrario que sera la bobina D y reservaremos.
2. Después de conectar el A, conectamos uno a uno los otros dos y uno deberá mover en el sentido de las manecillas del reloj, este sera el B, por lo tanto el otro sera el C.
3. Conectamos el C y si todo fue bien girara

2.Motores paso a paso

un paso en el sentido de las manecillas y el que apartamos D lo ponemos después y también.

Con los terminales localizados los conectamos a nuestro circuito y probamos.

3. LIBRERIA MPAP

Una vez tenemos conocimientos de lo que controlara esta librería y como se conectaría a nuestro arduino pasamos a explicar como utilizarla.

3.1. Constructores

La librería MpaP es una clase que consta de dos constructores para poder crearla:

MPaP(int, int, int, int):

Los parámetros son los pines donde conectaremos las entradas del Driver, de las bobinas A-B-C-D.

Además de definir los pines de salida, se definirán los parámetros por defecto:

- *Secuencia(secuencia)* de movimiento a NORMAL.
- Pasos máximos por segundo que aguanta el motor en 200 /pps).
- Ángulo por paso 18° (*angulo*)

3.Libreria MpaP

- Pasos necesarios para una vuelta en 20 (*pvuelta*)
- La máxima revoluciones por minuto que se calculan según los paso máximos por segundo y los pasos por vuelta siguen la formula $pps*60/pvuelta$ y es la velocidad máxima a alcanzar.
- Las revoluciones por minuto de velocidad del motor (*rpm*) se pondrá por defecto en el máximo.

MPaP(int, int, int, int, int):

Igual que el anterior pero ademas se define el número de pasos por vuelta. Con este constructor ya se define como es lógico el angulo por paso.

Los valores por defecto son los mismos excepto *pvuelta* que lo define el ultimo parámetro del constructor y el ángulo por paso que se define por la formula $360/pvuelta$.

Ejemplo de declaración

Libreria Motor pasos a paso (MPaP) UNIPOLARES

3.Libreria MpaP

MPaP M1(4, 7, 8, 11, 20);

creara un objeto M1 que gestionara el motor conectado a los pines digitales D4, D7, D8 y D11 y define que es un motor que necesita 20 pasos para dar una vuelta completa.

3.2. Variables

A estas variables accederemos con los métodos `get_<variable>` y la cambiaremos con `put_<variable>` de forma que tendremos:

- **`put_secuencia(int) get_secuencia()`**

Indica la secuencia en la que avanzara el motor y podrá ser cualquiera de los siguientes valores:

- ***NORMAL*** Avanza un paso con dos bobinas activas, entre las dos activadas.
- ***WAVE*** Avanza un paso con una bobina.
- ***MEDIO*** Avanza medio paso, algunos con dos bobinas y otros con una, con

3.Libreria MpaP

esta forma necesitaremos el doble de pasos para completar una vuelta, pero a su vez es más preciso el movimiento.

- ***put_pasos(int) int get_pasos()***

Pasos que se avanzan o retroceden ante el método avanza.

- ***float put_rpm(float) float put_rpm()
float get_rpm()***

Son las revoluciones por minuto a las que gira el motor, en este caso *put* devuelve las revoluciones puestas y en caso de ser mayor a las máximas se podrá esta. El método *put* sin parámetro pone las máxima revoluciones que aguanta el motor.

- ***put_pps(int) int get_pps()***

Son los máximos pasos por segundo que aguanta el motor, esta variable modifica la velocidad máxima del motor.

- ***put_angulo(float) float get_angulo()***

Ángulo por paso que realiza el motor, afecta a *pvuelta*.

3.Libreria MpaP

- ***put_pvuelta(int) int get_pvuelta()***

Pasos necesarios para completar una vuelta, afecta al ángulo por paso.

3.3. Métodos

Los métodos nos permitirá realizar acciones sobre nuestro motor y son:

- ***avanza()***

Realiza tantos pasos de motor en sentido horario, como indique la variable *pasos*.

- ***avanza(int)***

Realiza tantos pasos en sentido horario como se indican en el parámetro.

- ***retrocede()***

Realiza tantos pasos en sentido contrario a las agujas del reloj como indica la variable *pasos*.

- ***retrocede(int)***

Realiza los pasos indicados en el parámetro en el sentido contrario a las agujas del reloj.

3.Libreria MpaP

- ***angulo_a(float)***

Avanza en el sentido de las agujas del reloj, el ángulo indicado en el parámetro, con la precisión que nos permita el motor y esta definida en la variable *ángulo*.

- ***angulo_r(float)***

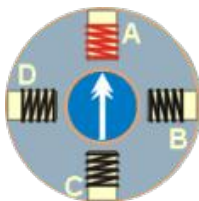
Retrocede el ángulo indicado en el parámetro.

4. Características motor

4. CARACTERÍSTICAS MOTOR

4.1. Unipolares

Los polos están colocados



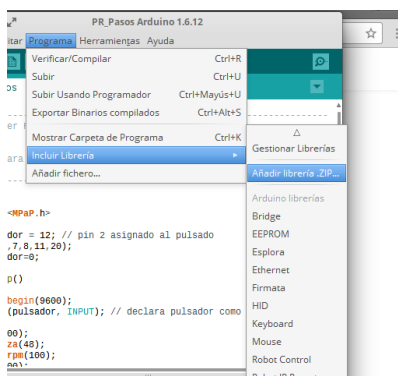
Modelo	Eléctrico	Cables	Paso/grados vuelta	Máx pasos
PM20S-020-HPJ4	12V, 100Ω, Bipolar 300 mA y 8Ω	6 cables A(BD), B, D, (AC)1, C	20 = 18°	500
PM35S-048-HPE8	24V, 35Ω, Bipolar 600 mA 7Ω	5 cables Común(3)A(1)B (4) C(5) D(2)	48 = 7,5°	200
PM35L-048-HPD4	24V, 30Ω Bipolar 500mA 5Ω	5 cables Comun(3)A(1)B (2)C(4)D(5)	48 = 7,5°	200

5.Instalación

5. INSTALACIÓN

Descargaremos el fichero “MpaP_lib.zip” de la URL

“http://www.pinguytaz.net/M_Archivos/MPaP_lib.zip” .



Abrimos el IDE de arduino y vamos a la opción “Programa→Incluir librería→añadir librería.-zip”, vamos a la ruta donde tenemos el fichero .zip y lo seleccionamos.

Si ahora cerramos el IDE y lo volvemos a arrancar podremos cargar los ejemplos de la

Librería Motor pasos a paso (MPaP) UNIPOLARES

5.Instalación

librería, desde archivos ejemplos ya que ahora aparecerá nuestra librería.

6. **GLOSARIO**

- **PWM** (Modulación de anchura de pulsos) nos permite lograr un resultado analógico mediante impulsos (activación y desactivación).